

**UNITED STATES PATENT APPLICATION**

**OF**

**John M. MARYNOWSKI,**

**Catalin D. VOINESCU,**

**AND**

**Stefan PUSCASU**

**FOR**

**AN AUTOMATED TRADING SYSTEM IN AN  
ELECTRONIC TRADING EXCHANGE**

048289-5003

## **BACKGROUND OF THE INVENTION**

### **Field of the Invention**

5       The present invention relates to an automated trading system for use in an electronic trading exchange network system and, more particularly, a trading system that rapidly, accurately, and safely responds to desirable trading opportunities.

### **Description of the Related Art**

Trading exchanges historically provided a location for buyers and sellers to meet to trade stocks, bonds, currencies, commodities, and other items. The New York Stock Exchange and the Chicago Mercantile Exchange are examples of such trading exchanges. Recent advances in computer and communications technology have led to electronic trading exchange system networks. Electronic trading exchange system networks use communications networks and computers to replicate traditional face-to-face exchange functions. For example, centralized exchange computers disseminate market information, maintain records and statistics, settle cash payments, determine risk based margin requirements, and match trades. Matching of trades is typically done on a first come-first served basis, whereby time of order entry is an important criterion for determining priority in fulfillment of a transaction.

A communications network connects the exchange computers to numerous trader sites.

20      Each trader site includes one or more trader stations operated by traders. Exchange network operators typically provide traders with interface software and, in some cases, hardware to enable traders to view prices and other information relating to products and to execute transactions by submitting orders and quotes. Orders are requests to buy or sell a specified amount of a particular item at a specified price. As an example, a trader may place an order to buy 100 shares of IBM stock for a bid price of 57.25. Quotes differ from orders in that quotes indicate bid price and ask prices, and bid and ask quantities for a particular item. As an example, a trader may place a quote indicating she is willing to buy 100 shares of IBM stock for a bid price of 57.0 and sell 100 shares of IBM stock at a price of 57.5. Retail customers, who may use brokers to execute their trades, typically will place an order to initiate a transaction. Professional

market makers can use either orders or quotes, but tend to fulfill their role of providing liquidity to a particular market by using quotes. This trading information is displayed in a grid or other organized format. Market competition is fierce. Traders who can quickly identify opportunities and act on them generate the largest profits.

5 Many trader stations in use today rely upon the traders themselves to decide whether to submit an order in response to a trading opportunity presented through the exchange. In this regard, the trading information is received from the exchange, processed, and displayed on a monitor of the trader's station. The trader reads the trading information from the monitor and decides whether or not to submit a matching order. The trader submits an order by entering  
10 instructions into the trader station using a keyboard or mouse, or even a gamepad or voice-activation as described in U.S. Application No. 08/273,362, filed on March 22, 1999 and incorporated herein by reference.

Attempts have been made to implement trading systems that automate decision-making so that orders may be submitted with limited trader interaction. These systems have a number of drawbacks. For example, user-friendly systems that automatically submit orders without trader interaction, while faster than a human trader, are believed to be relatively slow in terms of computer speed due to application and system design. In a typical set-up, trading information received from the exchange is processed by general purpose backend computer equipment. The backend computer may, among other things, (1) act as a gateway by communicating market information from the exchange to various types of client equipment, (2) submit, delete, and modify orders and quotes to the exchange from the various client equipment, (3) receive real-time trade confirmations and end-of-day back office reports, and (4) perform risk analysis, position management, and accounting functions. The trader stations are clients of the backend computer. The trader stations may be tasked with numerous functions, such as (1) receiving and displaying real-time market information, (2) creating and displaying theoretical prices related to market products, (3) composing, submitting, modifying, and deleting orders and quotes, and (4) maintaining positions and calculating risk management, to name a few. Each trader station is typically configured in a very user-friendly, Windows-based environment since the trader will spend long periods of time each day watching and interacting with it. The overhead associated  
15  
20  
25

with the functions performed by the backend computer and the trader stations reduces the response speed of automated trading.

In addition, computer equipment lacks the trading judgment of a human trader. A computer can generate staggering losses in the blink of an eye by submitting orders based upon incomplete or mistaken assumptions inherent in the trading program, erroneous input data, or corrupted data relied upon by the trading program. Accordingly, there exists a need in the art for an automated trading system that rapidly responds to trade information transmitted from an exchange, yet is safe and accurate.

10

### SUMMARY OF THE INVENTION

The present invention has been made in view of the above circumstances and has as an object to provide an improved trading system that rapidly responds to trading information received from an exchange.

A further object of the invention is to provide an automated trading system in an electronic trading exchange system that rapidly submits orders in response to trading information received from the exchange.

A further object of the invention is to provide an automated trading system that ensures the accuracy of automatic trading operations.

A further object of the invention is to provide an automated trading system that performs automatic trading operations without the risk of enormous losses due to erroneous, mistaken and/or repeated operation.

A further object of the invention is to provide a trading system in an automated trader system that may be remotely controlled.

A further object of the invention is to provide an automated trader system that automatically hedges some or all of the delta risk associated with the execution of a trade by submitting an order in connection with another, related trade opportunity.

Additional objects and advantages of the invention will be set forth in part in the description that follows, and in part will be obvious from the description, or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended

claims.

It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

5                   **BRIEF DESCRIPTION OF THE DRAWINGS**

The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiment(s) of the invention and together with the description, serve to explain the principles of the invention.

Figure 1 illustrates an embodiment of an electronic trading exchange system network in accordance with the present invention.

Figure 2 illustrates a further embodiment of an electronic trading system network in accordance with the present invention.

Figure 3 provides a schematic of the functionality of an embodiment of an automated trading system in accordance with the present invention.

Figure 4 illustrates an embodiment of a trading screen for use in connection with a trader station in accordance with the present invention.

Figure 5 provides a flow diagram of the steps performed in automated trading in accordance with the present invention.

Figure 6 illustrates a further embodiment of an electronic exchange system network in accordance with the present invention.

20                   **DESCRIPTION OF THE PREFERRED EMBODIMENT**

The present invention recognizes that electronic trading exchange system computers may match bid and ask prices on a first come/first serve basis. Accordingly, the speed and accuracy 25 of submitting orders or other responses is critical to the trader's ability to participate in the most profitable transactions. Even short delays in response may freeze a trader out of an otherwise lucrative transaction.

The present invention is capable of reducing the time it takes for the trader to submit an order or quote in response to incoming trading information from the exchange. In accordance 30 with one aspect of the present invention, the automated trading system automatically decides

whether or not to submit an order or quote based on calculation logic and decision logic, and trading information received from the exchange computers. To decrease the response time, the automated trading system may be dedicated or substantially dedicated to performing automated trading operations, with limited or minimized overhead permitted for other tasks unrelated to 5 trading. The present invention is further capable of reducing the time delay due to network lags arising from the transfer of trading information from the exchange computers to the automated trading system, and vice versa.

In an additional aspect of the present invention, safe and accurate automated trading may be achieved by performing various checks of the information used in decision-making and/or 10 information concerning the order. Further, an automated hedging feature may be invoked which, when a trader takes a position in a security, reduces the time needed to establish a delta hedge position in a related security.

Reference will now be made in detail to the present exemplary embodiment(s) of the 15 invention illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Figure 1 provides a schematic of an embodiment of an electronic trading exchange system network 10 that may be used in connection with the present invention. Other network arrangements may be used as well. The electronic trading exchange system network 10 includes an exchange site 100 and a plurality of trading sites 200. For purposes of simplification, Figure 20 1 illustrates an exchange site 100 linked to a single trading site 200. Other trading sites 200 may be located in a different part of the same city as the exchange site 100, a different city, a different country, or different continent as the exchange site 100. The exchange site 100 need not be limited to equipment provided at a single location, but may be provided in multiple locations linked by a network. Similarly, the trading sites 200 need not be limited to equipment provided 25 at a single location, but may include equipment at multiple locations linked by a network, such as a wide area network (WAN).

The exchange site 100 may be linked to the trading site 200 by one or more communication links 300. The communication links 300 may be part of a wide area network formed by dedicated communications lines, commonly-accessible communication lines, or a 30 combination thereof. For example, dedicated lines may be strung between the exchange site 100

and one or more of the member trading sites 200. Alternatively, dedicated lines may be leased from telephone, cable, or other communication network operators. For example, the public switched telephone network may embody the commonly-accessible communication lines. Of course, the communications links 300 may also include, in whole or in part, wireless communications, such as microwave or satellite links. While not shown in Figure 1, an intermediary (such as broker for a retail customer, or a clearing member for a professional market maker) may exist between the trading sites 200 and the exchange site 100. The role of an intermediary may include some level of risk analysis to confirm, say, sufficient capital exists to cover margin requirements for a potential resulting position, or some level of network routing to, say, ensure an order is sent to the intended exchange. Under the cases where risk analysis checks are passed and the routing information is proper, an intermediary should be transparent or substantially transparent to the functionality described herein and may be subsumed in the communication links 300. Of course, additional time delays may be introduced by any such intermediary processing.

In one embodiment, the exchange site 100 may be designed as a local area network (LAN) and include, for example, one or more security routers and one or more back office computers, among other equipment. For purposes of illustration only, two security routers 111, 112 and three back office computers 130-1, 130-2, 130-3 (referred to collectively as back office computer 130) are shown in Figure 1. The security routers 111, 112 control communications between the back office computers 130 and the communications links 300. Each security router 111, 112 transmits and receives communications over the communications links 300, as well as restricts communications from unauthorized sources. More particularly, the security router 111, 112 may be used to isolate the equipment at the exchange site 100 from intrusion and facilitate communication with the back office computers 130.

The back office computers 130 manage the trading of the various securities (e.g., futures, options, swaps or other derivatives; currencies, stocks, bonds, or other physicals like corn, precious metals, electricity, etc.) and/or other items traded on the exchange. For example, one or more of the back office computers 130 may function as market servers. In this capacity, they may maintain order books, perform order matching, generate market information for use at the exchange site 100 and/or for transmission over the communication links 300, and supply trade

information to other back office computers 130 for accounting and/or cash settlement purposes. One or more of the back office computers 130 may function as short-term accounting servers. As such, these computers may receive information from the market servers and generate information for transmission over the communication links 300. The short-term accounting servers may be initialized with status information from the previous day's trading before performing accounting tasks for the current day. One or more of the back office computers 130 may function as long-term account servers and, accordingly, function to collect data from the short-term accounting servers for batch processing and record-keeping. The long-term account servers may supply information to initialize the short-term account servers and generate reports for transmission to trading sites 200. Of course, the back office computers 130 may perform additional functions and a single computer may perform more than one of the above functions.

The trading sites 200 may include a LAN architecture having one or more security routers, one or more backend computers, one or more trader stations, and one or more hubs, among other equipment. For purposes of illustration only, Figure 1 shows two security routers 211, 212, two backend computers 220, 225, three trader stations 230-1, 230-2, 230-3, (collectively referred to as trader stations 230) and two hubs 240, 241. The security routers 211, 212 transfer trading information between the trading site 200 and the exchange site 100 and screen communications from unauthorized sources. The hubs 240, 241 distribute data between the backend computers 220, 225 and the trading stations 230.

Backend computer 220 may be configured as a communication server for the trader stations 230. The exchange may supply software and/or hardware for the backend computer 220 to facilitate communications with the exchange site 100. Alternatively, the exchange may simply provide protocol specifications that enable the trader site computers to communicate with the exchange computers. In this case, the trader has flexibility in choosing how it implements the protocol specifications running locally at trading site 200. Alternatively, if an intermediary site, such as a broker or clearing member site, is part of the communications routing between the exchange site and the trader site, the backend computer 220 may use communication software supplied by the intermediary site or, if permitted, communication software developed by the trader or a third party. Backend computer 220 handles communications between the trader

stations 230 and the back office computers 130 of the exchange. Of course, the trader site 200 may include multiple backend computers 220.

Backend computer 225 may also be equipped with software and/or hardware that facilitates communications with the exchange site and/or the intermediary site. Some exchanges, 5 for example, such as the EUREX (the German and Swiss Derivatives Exchange), recommend installation of a redundant on-site backend computer in the event that the primary communication backend computer 220 fails. In the case of the EUREX exchange, backend computer 225 may be configured to perform automated trading functions under the control of one or more of the trader stations 230. The automated trading functions are described in more 10 detail below. The backend computer 225 should be equipped with a high-speed processor and sufficient memory to efficiently handle automated trade processing. The trader stations 230 may control backend computer 225 remotely through a communication link 250, for example, a WAN. The trader site 200 may include multiple backend computers 225.

In one preferred embodiment, backend computer 225 is dedicated or substantially 15 dedicated to performing automated trading-related functions, as discussed in greater detail below. Backend computer 220, rather than backend computer 225, may be assigned trading-related tasks, such as (1) serving as a gateway to communicate market information from the exchange site 100 to trader stations 230, (2) submitting, deleting, and modifying orders and quotes to exchange site 100 from the trader stations 230, (3) receiving real-time trade 20 confirmations and end-of-day back office reports, and/or (4) performing risk analysis, position management, and accounting functions. In this way, backend computer 225 may perform automated trading functions with limited interruption or delays associated with other tasks the 25 backend computers (such as backend computer 220) may be requested to perform. This increases the response speed for automated trading operations. Moreover, the total time delay in submitting an order to the exchange site 100 includes a component attributable to the transmission delay or network lag in transmitting messages between the exchange site 100 and the trader site 200. Therefore, backend computer 225 is preferably located near to the equipment of the exchange site 100 to reduce delays associated with transmitting information and orders between the backend computer 225 and the exchange site 100. Accordingly, the total time for responding 30 to trading opportunities can be reduced both by reducing transmission delays and by increasing

decision-making speed at the trader site 200. Significantly, the backend computer 225 may be remotely supported or controlled by a distant trader station 230, which permits the trader station 230 to be located virtually anywhere without adversely affecting the response time of the automated trading system. Accordingly, the trader site 230 may be chosen based on 5 considerations such as tax, real estate costs, and quality of life, without having to worry that trader station location will impair the performance of automated trading carried on by backend computer 225.

Trader stations 230 receive information from the exchange site 100, process that information, and display at least part of it on a monitor. Each trader station 230 typically runs 10 interactive software that enables a trader, among other things, to submit orders and/or quotes to the exchange site 100. As discussed further below, one or more of the trader stations 230 may additionally be equipped with software for controlling the automated trading functions of backend computer 225.

Figure 2 illustrates an alternative embodiment of an electronic trading exchange system network 20. For the sake of brevity, features of Figure 2 similar to those in Figure 1, which are described above, will not be repeated. In Figure 2, the trading site 200 includes an automated trading system computer 225-2 separate from the backend computer 225-1. In this embodiment, the automated trading system computer 225-2 performs automated trading system functions and the backend computer 225-1 manages communications between the automated trading system computer 225-2 and the exchange site 100. The automated trading system computer 225-2 may be connected to the backend computer 225-1 using, for example, network interface cards or through a hub (not shown). The automated trading system computer 225-2 may be controlled using one or more trader stations 230 either locally or through a communication link 250. Alternatively, the automated trading system computer 225-2 may be controlled through the 25 communication computer 225-1 (as indicated by the dotted lines), which would serve to communicate information between the trader stations 230 and the automated trading system computer 225-2.

Figure 3 provides a functional diagram illustrating the operation of an embodiment of an automated trading system used in connection with options trading. Of course, the embodiment 30 may be modified for use in trading other securities (e.g., futures, swaps or other derivatives;

currencies, stocks, bonds, or other physicals like corn, precious metals, electricity, etc.) and/or other items traded on the exchange. As described in greater detail below, the general function of the automated trading system is to generate and automatically submit responses (such as orders or quotes) for transmission to an exchange based on market data received from the exchange and/or other sources. The automated trading system may perform this function in several different ways, as discussed greater detail below, including calculating a theoretical value for the traded item in real time when one or more of the underlying factors that affect the theoretical value changes. The calculated theoretical value is then used to determine whether an order or quote should be submitted based upon existing market prices for the traded item.

The automated trading system is preferably resident in the backend computer 225 as configured in Figure 1, which may utilize multiple CPUs. However, it may also be resident in one or more of the trader stations 230 or the backend computers 220. The automated trading system software may run in a text-based environment or a Windows or Windows-like environment. In some operating systems, automated trading may be assigned priority over other tasks or processes and run without debug messages.

The automated trading system receives and decodes current market information broadcast from the exchange site 100 through a receiver interface 410. The decoding of market information may be performed, for example, transparently by software supplied by the exchange for use with the backend computers, by the exchange software at the request or direction of the automated trading system, or by the automated trading system itself. The current market information may include information related to the options and an underlying security of the option. Market bid, ask and last prices and the day's volume for call and put options and the underlying security, to name a few, are typically received by the trader site. A call option is the right to buy the underlying security at a specified time in the future at a specified price, while a put option is the right to sell the underlying security at a specified time in the future at a specified price.

An option look-up table protocol 420 indexes an option data table 430. The option data table 430 stores information concerning options that may be automatically traded. For simplicity, a two-dimensional table having rows and columns will be described. However, it should be understood that higher-dimension arrays or tables may be used in connection with the

present invention. Each row of the option table 430 stores information relevant to a particular option including, for example, option name, current market option prices, times and quantities of the most recent trades by the trader, maximum order quantity, inputs that may needed to calculate option transaction prices, and whether automated trading is enabled for the option. As discussed further below, this information may be used as a check against erroneous operation.

5 Alternatively, option data table 430 may store information in connection with items that are actually being automatically traded at a given time. As a further alternative, option data table 430 may store information in connection with all of the options that may be subject to automated trading and include indices or pointers that link only the items currently enabled for automated

10 trading. Accordingly, any search of the option data table can skip those entries for which automated trading is not enabled. In such a case, an additional option data table may be maintained for the full set of items for which automated trading may be performed. This is useful in increasing the speed at which a disabled option can be enabled. Accuracy checks may use both the additional option data table and option data table 430. Communications between the automated trading system and the trader stations 230 are conducted through a trading station interface 440. For example, a trader station 230 may update information contained in the option data table 430 via a trading station interface 440. In this way, the option data table 430 may be updated to enable (disable) automated trading for a particular option.

The option data table 430 may be organized in several different ways. For example, the market bid and ask prices for a particular option may be stored in different rows of the option data table 430. Alternatively, the bid and ask prices may be stored in the same row of the option data table 430, but in different columns, or as different cells in a price dimension, for example. Also, the option data table 430 may be segmented, for example, so that all bid prices are grouped together and all ask prices are grouped together. Different classes of options (i.e., options with different underlying securities) may be indexed in a single data table 430 or in multiple look-up tables 430, for example, with each option having its own look-up table 430.

In addition to the current market information concerning option trading, the automated trading system may receive and decode current market information concerning the security (or securities) underlying the options. For example, an exchange that trades the underlying security typically maintains a book of bid (ask) prices and quantities of current order and quotes of those

traders wishing to buy (sell) the underlying security. The automated trading system may receive the underlying market information, for example, from the exchange site 100, from a separate exchange site, or from another market feed either directly or indirectly, e.g., through a trader station 230. The underlying market information for a given security may be stored in option data table 430, which may be formed in the memory of backend computer 225, since it may be one of the inputs needed to calculate option transaction prices. Theoretical price table 435, which may be formed in the memory of backend computer 225, may store option transaction prices for options defined in option data table 430 and/or the additional option data table described above.

The theoretical prices for derivatives, such as options, may be determined using complex mathematical models. Theoretical price logic 490 is tasked with generating the theoretical prices based on input information. Theoretical price logic 490 may be implemented in hardware or in a combination of hardware and software. For example, theoretical price logic may be implemented by the central processor and memory, and possible other equipment useful in performing fast mathematical calculations, in a general purpose computer. Alternatively, the theoretical price logic may be implemented in a separate processor in communication with the processor of a general purpose computer or an array of processors. Of course, theoretical price logic 490 may be embodied by other devices capable of generating theoretical prices as described herein.

Theoretical price logic 490 generates theoretical prices in accordance with mathematical models. The mathematical models produce a theoretical value for an option given values for a set of option pricing input variables that may change over time. Option pricing input variables considered in these models may include (1) the current market price of the underlying security (e.g., the price of the stock or future from which the option is derived), (2) interest rates, (3) the future volatility of underlying security, (4) dividend stream, (5) time until expiration, (6) whether the option can be exercised before the expiration date, and (7) whether the option is a call or put. Option pricing input variables (2)-(7) are not likely to change as frequently as the price of the underlying security, option pricing input variable (1). Some option pricing input variables, such as price of the underlying security, can be derived from the market. Other option pricing input variables, such as the future volatility of the underlying security, require some qualitative assessment by the trader.

The current market price of the underlying security may be defined in several different ways. At any given time during normal trading, the underlying security will usually have: (1) bid prices and quantities; (2) ask prices and quantities; (3) a last price and volume at which the underlying security was traded (last price); (4) an average of the current highest bid and lowest offer prices (average best bid, best ask price); and (5) an average price of a certain depth, among other values. The average price of a certain depth, say 5000 shares, would take the average of the: (a) best (highest) bid prices in the book of the first 5000 shares, and (b) best (lowest) offer prices in the book of the first 5000 shares. Obviously, there are many more definitions of underlying price that can be created, for example, using permutations of the five definitions provided above.

It is highly probable that at least four of these five definitions will yield (perhaps slightly) different results at any time. Since the normal hedging response of an option trade is to buy or sell the underlying security, the option trader may very carefully define underlying price used in her models. Specifically, buying (selling) calls and selling (buying) puts will usually lead to selling (buying) the underlying for hedging. For reasons discussed further below, the trader may want to set the theoretical buy price for call options and the theoretical sell price for put options using the bid price (and/or possibly bid underlying depth) of the underlying security. Likewise, the trader may want to set the theoretical sell price for call options and the theoretical buy price of put options using the ask price (and/or possibly the ask underlying depth) of the underlying security. In summary, theoretical value calculations used for automatic option trading may use any of several definitions of underlying price.

In addition to generating a theoretical value for an option, the trader may select a buy spread and a sell spread to compute an option transaction price. The buy spread may be subtracted from the theoretical value to produce the theoretical buy price – the highest price at which the trader is willing to buy a particular option using automated trading. The sell spread may be added to the theoretical value to produce the theoretical sell price – the lowest price at which the trader is willing to sell a particular option using automated trading. Accordingly, the trader would like to sell an option having a bid price that is the same as, or higher than, the trader's theoretical sell price. The trader would like to buy the option from anyone offering a price that is the same as, or lower than, the trader's theoretical buy price. Of course, buy and sell

spreads may be defined by the trader in either absolute terms (i.e. theoretical buy price = theoretical price - 0.10) or percentage terms (i.e. theoretical buy price = theoretical price - 0.01\* theoretical price). In summary, buy and sell spreads may be used to calculate option transaction prices such as theoretical buy and sell prices.

5 Accordingly, in the embodiment illustrated in Figure 3, the theoretical table 435 stores option transaction prices including a theoretical buy and sell price of the options for which automated trading is performed. For example, if automated trading is performed for options underlying Exxon stock, the theoretical price table 435 contains the theoretical buy and sell prices of Exxon stock options. If any of the inputs affecting option transaction price changes,  
10 the theoretical price table 435 can be updated. Of course, the theoretical price table 435 may be formed as part of the same table as option data table 430. However, option data table 430 and theoretical price table 435 will be described separately for the sake of convenience and not by way of limitation.

Similar to the option data table 430, the theoretical price table 435 may be organized in several ways. For example, all theoretical option buy prices for a given set of option pricing input variables (1)-(7) may be provided in a single column of theoretical price table 435, with a separate theoretical table provided for theoretical sell prices. Alternatively, the theoretical table 435 may index both a theoretical buy price and a theoretical sell price. The theoretical price table 435 may be segmented or multi-dimensional. Moreover, the theoretical price table 435 may be combined with, form a portion of, or be linked to option data table 430. The values in theoretical price table 435 may be compared with market option prices and may trigger automated buy or sell decisions by the system. While the theoretical price table 435 may take many forms, a table like the option data table 430 with each row representing a single option will be described for purposes of simplicity. As noted above, it should be understood that data structures other than tables may be used in connection with the present invention.  
25

In addition, the option data table 430 and theoretical price table 435 can be structured consistent with the particular search protocol used by the table update protocol 420 so that certain options or other items are located by the update protocol before other options or items. For example, if table update protocol 420 implements a linear search, the contents of option data table 430 for options at the top will be updated before options at the bottom even when  
30

multiple rows need updating. If the change in content affects the option transaction prices stored in theoretical price table 435, this implies options near the top could be recalculated and compared against market option prices before options at the bottom theoretical price table 435. Accordingly, the trader station 230 or the backend computer 225 may structure the option data 5 table 430 and/or the theoretical price table 435 so that options that have shown in the past, or are likely to show in the future, the most promising profits will be located first. The particular order of the options in the tables 430 and 435 may depend on the trading volume in an option, for example. Options with relatively high traded volumes over recent trading days or the current trading day may be given a higher priority ranking in table 430 and/or theoretical price 10 table 435. Moreover, some exchanges may limit the number of orders or quotes that a particular trader may have pending at a given time. For example, in some versions of the EUREX system, a trader can not submit a new quote before a confirmation that a previously submitted quote had been received from the exchange. Accordingly, structuring the tables 430 and 435 as described increases the opportunity for the trader to participate in the most lucrative transactions when there are restrictions on the number of concurrent orders placed. In addition, or in the alternative, the theoretical price logic 490 may calculate theoretical prices first for 15 options likely to yield the highest profits. The calculated theoretical prices may be supplied to the decision logic 490 either before the theoretical price logic calculates the theoretical price of another option or concurrently therewith.

In accordance with the embodiment shown in Figure 3, the trader station 230 may respond to changes in market conditions by changing any of the option pricing input variables 20 (1) - (7), changing buy or sell spreads, or changing any other variable that might effect option transaction prices. These changes would update the appropriate values for each option of the option data table 430 and then, since the changes affect option transaction prices, would trigger a recalculations of theoretical buy and/or sell prices in theoretical price table 435. Of course, updates to option data table 430 may originate directly from exchange 100 or backend 25 computer 220, or even some other source. For instance, option pricing input variable (1), the price of the underlying security, used in option pricing models, may be received dynamically by backend computer 225 from exchange 100 rather than being received from trading station

230. In this case, option data table 430, and, subsequently, theoretical price table 435 would be updated automatically in real time with no trader intervention.

Calculating the theoretical value for options or other financial derivatives can be relatively time consuming. The speed of these calculations becomes especially critical when 5 considering an exchange, such as the EUREX, may list some tens of thousands of options that a trader may want to continuously evaluate in real time. The time needed to calculate these values may depend upon the option's specifications, the particular mathematical model used to calculate the theoretical value, the use of pre-calculation or other calculation short cuts, and the level of desired precision of the calculated theoretical value. Obviously, hardware is also an important 10 consideration, as faster and more efficient computers tend to reduce calculation times. Below is a comparison of some average times (in microseconds, *μs*) needed to calculate a single theoretical price of some common options with different specifications using different mathematical models on PC using a Pentium III 450 MHz CPU and 128 MB of RAM:

Option Specifications			Models					
Type of Underlying Security	Early Exercise	Dividend	Black	Black - Scholes	Cox-Rubenstein	Garman-Kohlhagen	Implicit Finite Difference	Roll-Geske-Whaley
Future	No	No	1.9	1.9	32.5	NA	338	NA
Future	Yes	No	NA	NA	77.0	NA	397	NA
Stock	No	No	NA	1.7	32.2	NA	338	NA
Stock	Yes	No	NA	NA	76.7	NA	396	NA
Stock	No	Yes	NA	1.9	33.4	NA	NA	NA
Stock	Yes	Yes	NA	NA	78.0	NA	NA	67.7
Currency	No	No	NA	NA	32.5	NA	339	NA
Currency	Yes	No	NA	NA	80.7	2.1	397	NA

15

Note, an "NA" means the mathematical model is not appropriate to calculate a theoretical price of an option with those specifications. The option specifications and each of the theoretical price models are known in the art and will not be described here. Of course, other mathematical models may be used besides the ones listed above.

20

Note that these times can vary as much as a factor of about 200x. If speed were the only consideration for choosing the model to use for calculating values in theoretical price table 435,

then the models having the smallest computation times would probably be picked. However, other factors may be taken into account when selecting a model. For instance, a trader may find that certain models yield theoretical prices that are more market-realistic than other models. The trader will ultimately profit by trading (buying and selling) with other market participants. Thus, 5 a particular trader may select a model that, in the trader's opinion, best reflects these trading conditions, even at the expense of higher computation speeds. Accordingly, comparisons between theoretical prices and actual market prices may be necessary to decide which mathematical model to use. This comparison and model decision may be re-evaluated over time to reflect any changes in market conditions.

10 The times cited in the table above are typical for calculating a single theoretical option price without the use of pre-calculations, approximations, or other computational short cuts. Times for calculating subsequent theoretical prices for a given option may be reduced if part of the calculation made for the original option is stored and then used for the next calculation, or part of the calculation has been pre-calculated entirely before the first theoretical price has been calculated. Moreover, using part of the calculation of an option that is similar to the given option 15 may reduce the calculation time for a given option.

20 For instance, the Black and Black-Scholes models contain an exponential term ( $e^{-rt}$ ) related to the cost of carry. This term is unlikely to change frequently, since the three terms ( $e$  = natural log constant,  $r$  = the interest rate, and  $t$  = time until the expiration of the option) are unlikely to change frequently for a given trading day. This means that the term  $e^{-rt}$  may be pre-calculated before the first theoretical price is calculated, or stored after the first theoretical price 25 is calculated. Using the approach of pre-calculating the exponential term  $e^{-rt}$ , the computation times needed for the Black model for futures with no early exercise and the Black-Scholes for stock with no early exercise and a dividend can be reduced by about 30% (1.1  $\mu s$  vs 1.7  $\mu s$  for Black; 1.3  $\mu s$  versus 1.9  $\mu s$  for Black-Scholes model). Likewise, the Cox-Rubenstein model has calculations which are: (1) common to all strike prices for a given underlying price and volatility, and (2) common for a given strike price independent of underlying price. Using the approach of 30 using pre-calculated common values such as base nodes, the calculation times needed for the Cox-Rubenstein model for stock with dividend and early exercises may be reduced by about 10% (69.4  $\mu s$  versus 78.0  $\mu s$ ). Likewise, the Garman-Kohlhagen model has calculations that are

independent of strike price and underlying price. Specifically, there are two exponential terms like the one in Black and Black-Scholes, related to cost of carry in domestic and foreign currencies. If these values are pre-calculated, the computation times needed for the Garman-Kohlhagen model for currencies with early exercise can be reduced by about 40% (2.1 *us* versus 5 1.2 *us*). Likewise, the Roll-Geske-Whaley model uses a term that yields the underlying price assumed necessary for early exercise for call options. This term is independent of strike price and underlying price and is solved using numerical analysis. If the value of this term, as well as some exponential terms related to time decay are pre-calculated, the calculation times needed for 10 the Roll-Geske-Whaley model for stock with dividend and early exercises may be reduced by about 60% (37.2 *us* versus 67.7 *us*). Thus, pre-calculation of certain parts of the mathematical models used for determining theoretical prices may reduce total computation times of single or groups of similar options. Pre-calculation may be performed by back end computer 225, trader stations 230, back end computer 220, or another computational resource, for example, upon 15 initialization of these computers or upon selection of automated trading. Of course, rather than calculating the theoretical values in real time, a pre-calculated look-up table of theoretical values may be used, as described in U.S. Application No. 09/417,774 to Marynowksi et al., filed on October 14, 1999, and expressly incorporated herein by reference.

Pre-calculating certain parts of the mathematical models reduces computation times without compromising the precision of the models. This means a theoretical price of a given option under a given set of conditions calculated using a given model using pre-calculated values would be identical to the theoretical price calculated using no pre-calculated values. Also, pre-calculating certain parts of the mathematical models also allows the off-loading of this computational overhead from the computers assigned to automated decision making. For instance, in one preferred embodiment, the pre-calculating tasks can be moved from backend 20 computer 225 to, say, trading station 230. In this case, trading station 230 may calculate the required pre-calculated values upon initialization and any time one or more of the inputs of term being pre-calculated changed. The pre-calculated values may be stored in option data table 430 and used by backend computer 225 when updating theoretical price table 435.

Extrapolation may also be used to reduce computation times when using mathematical 25 models to compute theoretical option prices. Consider the case when a theoretical option price

has been computed for a given set of conditions. The input most likely to change is the price of the underlying security of the option. The mathematical models discussed above allow for the first derivative (the "delta" of an option) and second derivative (the "gamma" of an option) of the theoretical option price with respect underlying price to be calculated. If an option's theoretical 5 price, delta, and gamma are known for a given set of assumptions, and only the price of the underlying security changes, an extrapolated theoretical price at the new price of the underlying security may be calculated based on the previously calculated theoretical price and the delta and gamma. Consider the example where for given set of option pricing input variables (2) - (7) and price of the underlying security = 80.00, the theoretical price of the option = 7.20, delta = 0.40, 10 and gamma = 0.10. Now assume the price of the underlying security changes from 80.00 to 80.60, or a change of + 0.60. Using the extrapolation method, the new theoretical option price can be estimated as: Old Theoretical Option Price + (Change in Underlying Price) \* (Delta + Change in Underlying Price \* Gamma / 2) = 7.20 + (0.60) \* (0.40 + 0.60 \* 0.10 / 2) = 7.46. The 15 pre-calculated delta and gamma values may be stored in option data table 430, or elsewhere. The table below summarizes typical computation times (in microseconds) using baseline, pre-calculation and extrapolation methods:

Option Specifications			Model	Methodology		
Security	Early Exercise	Dividend		Baseline	Pre-Calculated	Extrapolation
Future	No	No	Black	1.9	1.1	0.48
Stock	No	Yes	Black-Scholes	1.9	1.3	0.48
Stock	Yes	Yes	Cox-Rubenstein	78.0	69.4	0.15
Currency	Yes	No	Garman-Kohlhagen	2.1	1.1	0.47
Stock	Yes	No	Implicit Finite Difference	396	NA	0.19
Stock	Yes	Yes	Roll-Geske-Whaley	67.7	37.2	0.90

Note that the extrapolation methodology times include the steps of calculating the delta and gamma at the original underlying security price, as well as the actual extrapolation step. As 20 the results above show, the extrapolation method yields considerably lower computation times compared to the baseline and pre-calculate methods. Also, note that the computation times of the extrapolation methodology are relatively consistent in absolute terms (i.e. less than 1.0 microsecond) across model type. This arises because the calculation of the delta and gamma, and the extrapolation step can be performed quickly relative to the entire process of calculating 25 the theoretical option price using each of the models. Of course, the delta and gamma may be

calculated in a prior step before the extrapolation calculation is initiated, which reduces the overall time to obtain the extrapolated theoretical price.

The advantage in speed of the extrapolation method may be offset by deficiencies in precision under certain circumstances. The extrapolation method assumes a constant gamma value over the range of the old and new underlying prices. This assumption may lead to differences in calculated theoretical option prices compared to using the baseline methodology with the same model at the new underlying price. The differences arise because the actual gamma was not constant across the range of underlying prices being considered. The magnitude of these differences may depend upon many factors, including time left to expiration, volatility assumed of the underlying security, strike price of the option relative to underlying security price, and magnitude of expected differences between the old and new underlying prices. Thus, the extrapolation method may not always be adopted, since there may be a trade-off between computation speed and precision. Accordingly, the trader may want to evaluate the magnitude of differences in precision for her particular trading conditions before adopting the extrapolation methodology.

Of course, the delta and gamma values used in the extrapolation method may be pre-calculated by a computer different than the computer making the automated trading decisions. For instance, in one preferred embodiment, the task of pre-calculating the delta and gamma values may be moved from backend computer 225 to, say, trading station 230. In this case, trading station 230 may calculate the required delta and gamma values upon initialization and any time one or more of the inputs used to calculate the delta and gamma values change by some pre-defined quantity. These delta and gamma values may be stored on backend computer 225 in option data table 430. If trading station 230 pre-calculates the delta and gamma values in option data table 430, the time for calculating a theoretical price using the extrapolation method becomes independent of the mathematical model used to generate the delta and gamma values stored in option data table 430. This is because the delta and gamma values are pre-calculated and do not have to be calculated as part of the extrapolation. Using the hardware specifications described above (i.e. 450 MHz Pentium III cpu, 128 MB RAM), typical computation times are less than 0.01 us.

Interpolation is another method to reduce computation time needed to calculate theoretical prices for options. Consider the case where the theoretical option prices are calculated for a given set of conditions, but for varying underlying prices. For instance, option pricing input variables (2) – (7) may be assumed constant and theoretical option prices may be calculated using, for example, underlying prices that vary the same amount above and below the current underlying price. Assume theoretical prices are calculated for a given set of conditions with option pricing input variables (2) – (7) remaining constant and for the following underlying prices to form an interpolation table:

Assumed Underlying Price	Theoretical Option Price
80.50	2.66
80.00	2.50
79.50	2.30

Using these points and assuming, say, a linear relationship between underlying price and option price, theoretical option prices may be calculated using interpolation for any new underlying price between 79.50 and 80.50. For instance, assume the underlying price changed to from 80.00 to 79.60. Using a linear interpolation method, the new theoretical option price can be calculated as:

$$\text{New Theo Price} = \text{Theo1} + (\text{AUPnew} - \text{AUP1}) * (\text{Theo2} - \text{Theo1}) / (\text{AUP2} - \text{AUP1})$$

where:

New Theo Price - the new theoretical option price for the new underlying price,  
Theo1 - the theoretical option price for the first assumed underlying price (2.30),  
AUPnew - the new underlying price (i.e., 79.60 in the example),  
AUP1 - the first assumed underlying price (79.5 in the example),  
Theo2 - the theoretical option price for the second assumed underlying price (2.50 in the example), and  
AUP2 - the second assumed underlying price (80.0).

In the example, the New Theo Price is calculated to be 2.34.

Interpolation is similar to extrapolation in that precision may be sacrificed since a constant, say linear, relationship is assumed to exist between theoretical option price and underlying price over the range of underlying prices used for interpolation. This constant relationship may not be valid and the actual theoretical option price calculated using the same model at, say 79.60, would be somewhat different than the interpolated result of 2.34. As with the extrapolation method, the magnitude of these differences may depend upon many factors. These factors include time left to expiration, assumed volatility of the underlying security, strike price of the option relative to underlying security price, difference between assumed underlying prices, and distance between the expected underlying prices and assumed underlying prices. Of course, there is no sacrifice in precision for any underlying price found in the interpolation table, since these theoretical option prices have been calculated exactly for those underlying prices.

Likewise, reducing the distance between assumed underlying prices used for interpolating may increase precision. For example, adding theoretical option prices for underlying prices of 80.25 and 79.75 to the interpolation table above would probably improve precision of calculated theoretical option prices. This arises because the (1) assumed linear relationship between underlying price and theoretical option price is over a smaller underlying price distance, and (2) chance of an underlying price matching an underlying price already in the table increases. Of course, if the new underlying price equals one of the assumed underlying prices in the interpolation table, no interpolation is required and the theoretical prices corresponding to the underlying price may be passed to the theoretical price table 435. Obviously, the interpolation table can be expanded to include all possible underlying prices within an expected underlying range, since the exchange usually defines a minimum "tick size" (i.e., smallest increment of change) of the underlying security. In the limit of the interpolation table including all possible underlying prices (say between 79.00 and 81.00, in steps of 0.01) the interpolation table resembles a look-up table since no interpolation is necessary for any underlying price within the underlying price range. The use of a look-up table to calculate theoretical values is described in U.S. Application No. 09/417,774 to Marynowski et al., which is incorporated by reference.

Times needed to calculate new theoretical option prices are the fastest when values in the table above are pre-calculated and the new underlying price falls in the range of the underlying

prices in the table. Under these conditions and using the hardware specifications described above (i.e., 450 MHz Pentium III cpu, 128 MB RAM), typical computation times for underlying prices not matching an underlying price in the table are less than 0.01 us. Interpolation between any two assumed underlying prices requires a constant time (i.e. less than 0.01 us), regardless of model used to calculate the theoretical prices if the theoretical prices are in the pre-calculated interpolation table. Calculation of the theoretical option prices can be done away from backend computer 225 on, say, trading station 230. The interpolation table may be dynamically centered around the market underlying price. For instance, trading station 230 may be asked to calculate and send backend computer 225 a new interpolation table if the underlying price moves outside of some pre-defined underlying price range. The interpolation table, containing underlying prices and the corresponding theoretical option prices, can be stored in option data table 430 on backend computer 225, in the theoretical price logic, or in another accessible memory location. New theoretical option prices calculated for the current underlying price may be stored in theoretical price table 435.

Referring still to Figure 3, decision logic 450 compares the theoretical price calculated and stored in the theoretical price table 435 to the market price for the option and, based on the comparison, determines whether the option should be bought or sold, or no action should be taken. For example, in an embodiment in which the theoretical price table 435 stores theoretical buy and sell prices for a particular option, decisions may be triggered when: (1) a theoretical option price in table 435 changes, but the market bid and ask prices of the option remain the same, (2) the market bid or ask price of the option changes, but the theoretical prices in table 435 remains the same, (3) automated trading is enabled for a particular option, and (4) when safety checks are relaxed for a particular option.

Consider example (1) in which a theoretical price of a particular option stored in theoretical price table 435 changes and the bid and ask prices of an option remain static. As noted above, the theoretical price table 435 may be updated when one or more of the values that affect the theoretical buy and sell prices changes such as, but not limited to, the buy and ask spreads and/or option pricing input variables (1) - (7). For example, option pricing input variables (1) - (7) discussed above could change, perhaps due to a change in the trader's assessment of market conditions. These changes may occur when the trader enters new

information through a trader station 230 or when new information becomes available through another source (e.g., a change in interest rate occurs in a database associated with the trading site 200). A change in one or more of option pricing input variables (1) - (7) triggers a re-computation of (probably) all values in the theoretical price table 435. As noted above, re-  
5 computation may involve re-calculating the theoretical price in its entirety or using pre-calculated values, extrapolation, and/or interpolation. For purposes of discussion below, assume option pricing input variable (1), the price of the underlying security, changes and thereby changes a theoretical buy or sell price of a particular option in theoretical price table 435. Decision logic 450 will compare the current market ask (bid) price of the option to the new  
10 theoretical buy (sell) price obtained from the theoretical price table 435. In this case, the decision logic 450 performs all comparisons affected by the change in underlying price. For example, a change in the bid (ask) price of the underlying security may affect the theoretical buy (sell) price of some or all call options and the theoretical sell (buy) price of some or all put options associated with the underlying security. Accordingly, the decision logic 450 makes comparisons of market bid or ask prices corresponding to new theoretical sell and buy prices.  
15

Consider example (2) in which the market bid (ask) price for a particular option changes and the theoretical prices of the option remain constant. The decision logic 450 will compare the new market bid (ask) price to the corresponding theoretical sell (buy) price that exists at that time from the theoretical price table 435. Accordingly, a change in market bid (ask) price of a particular option may trigger a comparison of market bid (ask) price to theoretical sell (buy) price. Based on the comparison, for example, if the market bid (ask) price is greater (less) than or equal to the theoretical sell (buy) price, the automated trading system may prepare a response (such as an order or quote) for the particular option.  
20

Consider example (3) when automated buying or selling trading for a particular option is changed from disabled to an enabled state. This could arise, for instance, at the beginning of the trading day if the default state of a new trading session on trading station 230 is all options disabled. Enabling automated selling (buying) for a particular option or group of options can trigger decision logic 450 to make a comparison of the market bid (ask) prices to the theoretical sell (buy) price in table 435.  
25

In addition to enabled and disabled states, a third, "warming up" or "test" state may be provided for an option in the automated trading system. In this third state, the automated trading system may perform all steps except actually placing an order. This allows the trader to monitor the operation of the automated trading system without actually submitting orders, thereby 5 reducing the risk of enabling options for automatic trading using theoretical prices which are not market realistic.

Consider example (4) in which a safety check for a particular option is relaxed. This could arise, for example, if a global safety check condition implemented by safety check logic 460 is disabled or changed. For example, a safety check condition relating to the maximum 10 quantity or frequency of trading attempts of a particular option may be increased via a command from trading station 230. In connection with the trading frequency condition, the entire automated trading system may be held in a "pause" state if it had made more than a predetermined number (e.g., 3) automated trading attempts within a predetermined time period (e.g., 60 seconds). If this global safety check is disabled or relaxed, for example, by increasing the predetermined number of attempts (e.g., from 3 to 5), the trading frequency safety check may no longer be in violation. As a result, the entire automated trading system may transition from the "paused" state to the enabled state. If a particular option had been enabled for automated selling (buying), the decision logic 450 will then compare the market bid (ask) price to the theoretical sell (buy) price in table 435.

Of course, the automated trading system may be designed to automatically switch from the "enabled" to "paused" state if conditions are deemed too risky to run automated trading. For example, the automated trading system may change from an "enabled" state to a "paused" state when it senses, or receives a message, that: (1) communication in any of the communication links is not working properly, (2) trading has halted or closed in the underlying security and/or 20 options of a particular security, (3) the options for a particular security are trading in a "fast market", as determined by exchange officials, (4) the difference between the ask and bid price of the underlying security is greater than some predetermined value, (5) the rate of change of the price of the underlying security is greater than some predetermined value, (6) trading in the underlying security is in an "auction" or "crossing" state, rather than normal "bid" and "ask" 25 trading, and/or (7) release of known news events is pending. Assume the system automatically 30 trading, and/or (7) release of known news events is pending. Assume the system automatically

went into the "paused" state due to one of the conditions above. The system can be designed to either automatically go into the "warm-up" state when the triggering condition has passed, or require manual intervention to move from the "paused" to the "warm-up" stage. Obviously, the system should be designed to also let the trader manually switch the automated trading system 5 from "enabled" to the "paused" state whenever the trader desires.

Decision logic 450 determines that a sell (buy) order should be submitted if the market bid (ask) price is greater (less) than or equal to the theoretical sell (buy) price. Even if decision logic 450 determines that an order should be submitted, safety check logic 460 may be used to prevent an order from being submitted. Safety check logic 460, for example, can block orders 10 entirely, or put a cap on the maximum quantity attempted to be bought or sold, for an option when acceptance of that order would result in the trader having a position greater than a predetermined threshold quantity of that option. A trader may be also set a limit on, say, the total delta of an automated trading attempt. This may arise if she is concerned about the potential size of the delta hedge that might be needed relative to the depth and liquidity of the underlying security market. This would mean that each option might have a different maximum 15 quantity and the quantity would be inversely proportional to the delta of the option. Also, the automated trading system may be paused or stopped if the number of attempted orders exceeds a predetermined amount in a predetermined period of time. The constraints may be stored in option data table 430 or elsewhere and may be varied for individual options. Other constraints 20 may involve generating warnings and/or preventing orders, for example, when: (1) the theoretical buy price exceeds the theoretical ask price, (2) the theoretical buy price exceeds the theoretical value, (3) the theoretical sell price is less than the theoretical value, (4) the price of the underlying security moves outside some range, and/or (5) the theoretical sell price is less than the intrinsic value of the option. The intrinsic value may be defined as the difference between 25 the strike price and the market price of the underlying security for puts, and the difference between the market price of the underlying security and the strike price for calls, where the minimum intrinsic value is zero. The trader may be able to override some or all of the checks performed by safety check logic 460 to increase speed of automated trading.

If the safety checks are passed (or overridden), order logic 470 creates a response and 30 submits the response to the exchange site 100 via an output interface 480. The trading station

230 may be notified through a trading station interface whether or not the safety checks are passed. The output interface 480 may pass the order to exchange interface software for ultimate transmission to the exchange site 100. The receiver interface 410 and the output interface 480 may be formed by common equipment and/or data ports.

5       The option data table 430 and theoretical price table 435 can be checked periodically to ensure the accuracy of its content. For example, checks may be performed every, say, 15 seconds. This can be done, for example, by performing a checksum operation in which the entries of tables 430 and 435 table are summed and the sums are compared with the sums of a corresponding tables maintained by a trader station 230. If the sums match, option data table 430 and theoretical price table 435 may be presumed to be accurate. If the sums do not match, a warning is generated and automated trading is stopped completely or paused until option data table 430 and theoretical price table 435 are reloaded or updated and accuracy can be ensured. Of course, other or additional techniques for testing the accuracy of tables 430 and 435 may be implemented. Moreover, such an accuracy check may be omitted if one is sufficiently confident in the reliability of the software, hardware and communication networks.

10      Knowledge of how the search protocol locates data within the option data table 430 and theoretical price table 435 may be used to structure these tables to ensure that selected options will be located particularly quickly. The selected options may be, for example, frequently traded options and/or options whose price will become attractive with a small change in the underlying security price. For example, the search protocol may conduct searches by starting at the first row of the table and then stepping through each successive row until a particular row is identified. In this case, the tables 430 and 435 may be structured so that a select option is placed in the first row. Consequently, the search protocol will locate the select option first. Statistics may be maintained, for example, at a trader station 230, and used to restructure the tables 430 and 435 as 15 trading conditions change. Further, when the market price of the underlying security changes, the theoretical price logic 490 may calculate the new theoretical prices in the same predetermined order as the search protocol, with the newly calculated theoretical price acted upon by the decision logic 450 either before or during calculation of the next theoretical price. In this way, the automated trading system calculates theoretical value and makes transaction 20 decisions first for options believed to be most likely to generate profitable transactions, whether 25 30

the decision logic 450 is triggered by a change in market price of the option, by a change in theoretical value, or otherwise.

The embodiment described in connection with Figure 3 compares the current market price of an option to theoretical buy and sell prices from a theoretical price table 435 to make a buy/sell decision. However, other transaction values may be compared consistent with the present invention to generate buy/sell decisions. For example, the theoretical option value may be subtracted from the market bid (ask) price and compared to a sell (buy) spread selected by the trader to generate buy/sell decisions. Alternatively, implied volatilities may be calculated for market option bid (ask) prices using, say, mathematical models and inputs similar to those used for calculating theoretical option prices. These calculated implied volatilities may then be compared to trader-defined theoretical volatility values to make buy/sell decisions. Of course, other values may also be indexed and used for comparison to generate buy/sell decisions consistent with the present invention.

Figure 4 illustrates an embodiment of a trader screen 500 displayed on a trader station 230 in connection with trading options on a particular security or commodity. The trading screen 500 may provide a graphic user interface to enable the trader to set parameters associated with automated trading. Trading screen 500 is organized as an array of cells 510. The rows 512 of the array represent different options available in the market for the particular security or commodity. The columns 514 of the array provide information concerning the options. More particularly, the columns to the left of the "Strike" column provide information on call options and the columns to the right of the "Mon" column provide information on put options. Call and put options are, thus, displayed as mirror images of each other.

Each row of the array represents information relating to a different pair of call and put options for a particular strike price, month and year. The first column from left to right is labeled "DCX," which identifies the underlying security for the options as Daimler-Chrysler stock. The values below the "DCX" label consecutively number the rows of the array. The trading screen may be scrolled up or down to view additional rows in the array, if any exist. The next fourteen columns contain information relating to call options. The second column, "POS," is to the right of the "DCX" column. The values below the column heading POS indicate the trader's position (i.e., how many of the options the trader possesses) in call options for each row of the array. A

negative cell value in the “POS” column indicates that the trader has sold more of the option than she has bought (this is called a *short* position. Positive values denote a *long* position). Cells in the “B” column (three columns to the right of the “POS” column) indicate whether automated buying is enabled for the particular options corresponding to those cells. Cells in the “S” column (three columns to the right of the “B” column) indicate whether automated selling is enabled for the particular options corresponding to those cells. The trader may select one or more sells in the “B” and “S” columns to enable or disable automated buying and selling, respectively, of options corresponding to the selected cells.

The “TBid” and “TAsk” columns indicate the theoretical buy and sell prices for automated trading. The “Theo” column represents the theoretical value assigned to the call option for each row. To the right of the “Mon” column, the screen provides “TBid,” “TAsk,” “Theo,” and “POS” columns, among others, for the put options in each row of the array. Additional details concerning the remaining columns of the trader screen 500, as well as other information concerning its functionality, can be found in U.S. Application No. 09/273,362 to Marynowski et al., filed March 22, 1999, and U.S Application No. 09/417,774 to Marynowski et al., filed October 14, 1999, both of which are expressly incorporated herein by reference.

The “POS” columns usually provide information received from the exchange site and are not adjustable by the trader. The “TBid,” “TAsk,” and “Theo” columns may be adjusted by the trader using a mouse, keyboard, or other input device, such as a game pad. For example, the trader may select a particular “TBid” or “TAsk” cell by clicking once and then using up or down arrows, for instance, to increase (arrow up) or decrease (arrow down) the value. Mathematically, this may be achieved by increasing or decreasing the bid spread value (BSprd) or the ask spread value (ASprd). Changes made to the BSprd and ASprd values on trading station 230 would update option table data 430, and, consequently, may trigger a recalculation of theoretical price table 435. Note that changes to BSprd and ASprd values may not effect the “Theo” value since BSprd and ASprd are not inputs into the “Theo” calculation. A particular “Theo” cell may be adjusted in the same manner as a “TBid” or “TAsk” cell. Mathematically, adjustments to a Theo cell may be achieved by increasing or decreasing the assumed volatility of that particular option. Changes made to assumed volatilities on trading station 230 would update option table data 430, and, consequently, trigger a recalculation of theoretical price table 435. The “TBid”, “TAsk”,

and "Theo" values may also be adjustable in groups, for example, by selecting multiple cells or all cells in the column by selecting the column header. The trader station 230 may update the displayed values of Theo, TBid and TAsk values as the underlying security price change, or any variable contributing to Theo, TBid, or TAsk change (such as option pricing input variables (2) - 5 (7) discussed above). For example, the trader station 230 may receive a market feed providing price information concerning the underlying security. The price information may be used to update or refresh the trading screen 500. This may include the displayed TBid, TAsk, and Theo values for a given underlying price.

Figure 5 provides an exemplary progression of steps from transmission of the current 10 market information from the exchange site 100 to receipt of trade confirmation by the trader site 200 and the delay experienced at each step. The progression of steps illustrated assumes that the exchange-required interface software runs locally on backend computers 220 and 225. Link 1 represents the line delay experienced by current trading information as it is transmitted from the exchange site 100 to the trader site 200. Locating the automated trading system close to the exchange site 100 reduces the line delay of Link 1 (as well as that of Link 15). Thus, by 15 reducing the delay associated with making automated trading decisions as well as the associated line delay, the overall speed in submitting orders to the exchange site 100 is increased. Moreover, the trader station 230 that monitors and controls the backend computer 225 that implements the automated trading need not be located close to the exchange site 100, but may 20 monitor and control the backend computer 225 remotely.

Link 2 represents delays associated with operating system (networking subsystem) related to receiving data packets from the exchange site 100. One technique for reducing this delay is to choose a platform, such as VMS or Linux, that has a good quality implementation of networking services used in automated trading.

Link 3 represents delays associated with decompressing information received from the exchange site 100. Link 4 represents other processing delays that may be inherent in exchange interface software provided by the exchange for use at the trader site 200. The exchange interface software allows the trader's equipment to interface with the exchange equipment. The exchange may impose obligations requiring traders to use the exchange interface software in 25 trading. The exchange interface software, for example, may process the received market data 30

and supply the data to an interface of an automated trading application installed by the trader. For instance, the market data may be input to internal tables and/or may be converted to actual values. Links 5 and 6 represent delays associated with the distribution of information from the exchange interface software to an interface of the trading system application. The exchange site 5 100 typically broadcasts information concerning all traded items. Each trading application usually subscribes to several sources of data (e.g., market data and trade confirmations for several products). In some cases, the exchange interface software will receive and decode all information received from the exchange site 100, but only pass some of the information to the interface of the automated trading system. The exchange interface software spends some time in 10 determining whether a particular piece of market information should be passed to the automated trading system. The exchange interface software and the trading system interface software communicate via a protocol. For example, the exchange interface software may notify the automated trading system via a callback function supplied by the latter, or by some other operating-system dependent mechanism (e.g., mailboxes on VMS). Choosing a platform that 15 efficiently supports the protocol chosen can reduce this delay.

After receiving the current market information, the automated trading system decodes the information and, using a search protocol, searches a table of traded products, resulting in a delay indicated by link 7a. A hash table with an efficient hash key or a search tree may be used to reduce the delay associated with the processing associated with link 7a. The particular search 20 protocol should be fine-tuned to the platform used for the automated trading system as performance may vary with the platform to the extent that a linear search may prove better than a hash table even for a surprisingly large number of products (over 100). The look-up time for hash tables is almost constant. For binary trees, the look-up time is proportional to the logarithm of N (in  $O(\log N)$ ), where N is the number of products traded. A linear search has a look-up time 25 that is proportional to N (in  $O(N)$ ). Of course, the actual times encountered in practice matter, so the look-up protocol should be tailored to the platform used.

Link 7b represents the delay in updating the option data table 430 and recalculating table 435, if necessary. Current market information received in link 7a may include new prices related to the underlying security. A new underlying security price may trigger the underlying security 30 price values in option data table 430 to be updated. This, in turn, may trigger a recalculation of

values contained in the theoretical price table 435. Of course, current market information in link 7a may also contain other variables that may effect the theoretical price table 435. In such cases, similar steps of updating option data table 430 and recalculating theoretical price table 435 may be necessary. If current market information received in link 7a does not change any value in 5 option table 430 that effects theoretical price table 430, no actions are taken and the process continues to link 8, bypassing link 7b completely. Of course, values in option data table 430 may be updated from other sources besides exchange site 100. For example, trading station 230 may send a new assumed volatility for a particular option without backend computer 225 processing links 1-7a. In this case, option data table 430 and theoretical prices table 435 will be 10 updated in link 7b and the process continues to link 8.

Link 8 represents the delay attributable to decision-making and safety checks. As noted above, decisions are made based on a numeric comparison between the current market price and the corresponding theoretical price. Safety checks account for most of the delay experienced in link 8. Safety checks may include, among other things, (1) price and quantity reasonability checks, (2) trade attempt frequency limitations, and (3) underlying bid and ask price checks. 15

Links 9-15 corresponds to the delay associated with composing an order and submitting the order to the exchange. In particular, link 9 reflects the time spent composing the order, which may require a format defined by the exchange. Link 10 corresponds to the time required for the automated trading system output interface to communicate the order to the exchange interface software. This may be done, for example, using a synchronous function call or an asynchronous call. In some embodiments, the tasks associated with links 9 and 10 may be performed at the same time. Links 11 and 12 correspond to the time expended while the exchange interface software receives the order, decides which module should be used to submit the order, interprets the order request, and performs a series of validations. If the order passes 20 these tests, it is converted into the exchange format and passed to the exchange, as indicated by links 13-15. 25

Links 1 and 15 may include any delays occurring between the trader site 200 and the exchange site 100, including, for example, any delays attributable to an intermediary, such as a broker. As noted above, the delays attributable to links 1 and 15 may be reduced by locating the 30 automated trading system close to the exchange site 100. If, for example, the exchange site for

an option is located in a city distant from the exchange site for the underlying stock, the trader site may be located in between the two exchange sites to reduce transmission delays, or closer to the exchange site that is expected to trigger the most trading activity for the automated trading system and thus minimize the effect of transmission delays on automated trading. In addition, if routers and LANs are used at the trader site 200, the selection of high-speed equipment may reduce delays and/or priority schemes. The delay experienced in links 2-14 may be reduced, of course, by using a faster computer. However, the efficiency of the software and algorithms is also an important factor in reducing delay. Further, in some situations, it is possible to integrate the automated trading system software with the software that interfaces with the exchange site 100, which leads to reduced delay. In such a case, the automated trading system receiver and output interfaces may be the same as the exchange receiver and output interfaces.

Link 16 reflects the processing of the order at the exchange site 100. Following the exchange site 100 processing, a confirmation of the trade is returned to the trader if the trader's order is matched. Not all orders result in a match. There may be no sharing of lucrative trades with other traders who may have submitted similar matching orders that are received by the exchange even some microseconds later.

As noted above, the embodiment illustrated in Figure 5 corresponds to an arrangement in which the interface software provided by the exchange and the automated trading system are resident on the backend computer 225. In arrangements in which the interface software and the automated trading system are resident on separate backend computers, the vertical dashed line 610 indicates the interface between the separate computers. The separate backend computers may be connected via network interface cards or a common hub. Additional delays may be experienced in the transmission and reception of between the backend computers as well as from LAN throughput and latency.

In arrangements in which there is no exchange-provided interface software running locally on backend computers 220 and 225, steps 2-6 and 10-14 may be performed by the automated trading software or eliminated completely. For example, steps 10 and 11 may not be needed since they are usually associated with communication between two applications, and not communication within a single application. Similarly, the trader may be able to choose not to perform the processing and pre-order validation performed in step 12 when using the proprietary

trader system software. Whether or not compression per step 13 is performed depends on the communications protocols supported by the exchange. These protocols may be spelled out in an application programming interface (API) manual provided to the trader by the exchange.

Moreover, in arrangements in which an intermediary such as a broker or clearing agent is included in the communication routing between trader site 200 and exchange site 100, steps 2-6 and 10-14 may be performed by communication software provided by the intermediary, or software that uses API's provided by the intermediary and is developed by the trader or some third party vendor. In this case, the intermediary would thus be responsible for translating communications between its protocol and that of the exchange before routing messages between the exchange and the trader. This would of course be outside the control of the trader and would introduce additional delays in the total response.

Figure 6 provides a schematic of an embodiment of an electronic trading exchange system network 70 coupled to multiple trading sites. The electronic trading exchange system network 70 is similar to that shown as 10 in Figure 1 and, for the sake of brevity, duplicative description will be omitted.

As shown in Figure 6, exchange site 700 is coupled to trader site 200 by communication links 300. In one embodiment, the exchange site 700 may be designed as a local area network (LAN) and include, for example, one or more security routers and one or more back office computers, among other equipment. For purposes of illustration only, a single security router 710 and three back office computers 730-1, 730-2, 730-3 (referred to collectively as back office computers 730) are shown in Figure 6. Security router 710 controls communications between the back office computers 730 and the communications link 300. Security router 710 transmits and receives communications over the communications link 300, as well as restricts communications from unauthorized sources. More particularly, the security router 710 may be used to isolate the equipment at the exchange site 700 from intrusion and facilitate communication with the back office computers 730. The back office computers 730 manage the trading of the various securities, currencies, commodities and/or other items traded on the exchange. In this regard, back office computers 730 may function similarly to the back office computers 130 of exchange site 100.

For purposes of illustration only, trading site 200 additionally includes a security router

213 and a backend computer 223 coupled to hub 240. The security router 213 and backend computer 223 may be located remotely from other equipment of the trader site 200. Security router 213 transfers trading information between the trading site 200 and the exchange site 700. As above, the security router 213 screens communications from unauthorized sources.

5 Backend computer 223 may be configured as a communication server for the trader stations 230. Hub 240 handles communications between backend computer 223 and trader stations 230.

In the embodiment shown in Figure 6, trader site 200 is connected to a first exchange site 100 and a second exchange site 700. Of course, other network arrangements may be used 10 in connection with the present invention. Through the first exchange site 100, the trader site 200 may receive market information and trade securities, such as options, futures, and other derivatives; currencies, stocks, bonds, and other physicals like corn, metals, electricity, etc., and/or other items. Through the second exchange site 700, the trader site 200 may receive market information and/or trade securities and/or other items. Traders site 200 receives market 15 information and trades securities or other items related to the securities or other items traded through the first exchange site 100.

The trader site 200 may be equipped with hedging capability that facilitates buying or selling securities and/or other items traded on the exchange to hedge at least some of the risk (for example, delta risk) associated with an automated trade for other securities or items. For 20 example, trader site 200 may trade options for a particular stock through exchange site 100 and the particular stock through exchange site 700. In general, two types of orders may be submitted to an exchange to buy (sell) a security. A market order instructs the exchange to buy 25 (sell) a specified quantity of the security at the going market price. A limit order instructs the exchange to buy (sell) up to a specific quantity of the security if the market price is equal or better than a specified value. A trader usually can be assured that (under most circumstances) a market order will be filled by the exchange, but cannot be certain of the price at which the order is filled. The actual price that the market order is filled depends on available price and depth of market. While the trader placing a limit order can be assured of a price, all or a portion of the limit order may never be filled if the market price never meets the limit order 30 conditions.

The principles of a market order and a limit order are illustrated by the following examples.

<b>Market Order Price</b>	<b>Market Order Quantity to Sell</b>	<b>Bid Price</b>	<b>Bid Amount</b>	<b>Matched Against Order?</b>
Best Available Market Prices	1000 shares	\$110/share	400 shares	Yes
		\$100/share	600 shares	Yes
		\$80/share	2000 shares	No

5 The above table assumes a market with current bids of \$110/share for 400 shares, \$100/share for 600 shares and \$80/share for 2000 shares, as indicated above. A market order to sell 1000 shares will be executed by the exchange at an average price of \$104/share. In other words, the 1000 share market order will be matched with 400 shares at \$110/share and 600 shares at \$100/share, for a net of 1000 shares at an average price of \$104/share.

10 If the bid of \$110/share for 400 shares is sold just before the market order is received, the following market is presented.

<b>Market Order Price</b>	<b>Market Order Quantity to Sell</b>	<b>Bid Price</b>	<b>Bid Amount</b>	<b>Matched Against Order?</b>
Best Available Market Prices	1000 shares	\$110/share	400 shares	Not Available
		\$100/share	600 shares	Yes
		\$80/share	2000 shares	Yes

Because 400 shares at \$110/share is no longer available, the exchange will match the market order using 600 shares of the \$100/share bid and 400 shares of the \$80/share bid, resulting in an average price of \$92/share.

We next consider a similar scenario using limit orders instead of market orders.

Limit Order Price	Market Order Quantity to Sell	Bid Price	Bid Amount	Matched Against Order?
\$100/share	1000 shares	\$110/share	400 shares	Not Available
		\$100/share	600 shares	Yes
		\$80/share	2000 shares	No

Now assume that a limit order to sell 1000 shares at \$100/share is submitted instead of the market order and the \$110/share bid has already been matched. The exchange will match 600 shares of the limit order at \$100/share and will not match the remaining 400 shares because the \$80/share bid is too low. Accordingly, the remaining 400 shares of the limit order will stay in the exchange's book until a new matching order to buy at \$100/share or higher is received by the exchange, which may never occur, or until it is cancelled.

As discussed above, order submission in the automated trading system depends, for example, on the price of the underlying security, which is liable to change frequently. Thus, if the automated trading system makes an option trade, the trader may wish to hedge the risk associated with underlying price movement. This risk, commonly called delta risk, may be quantified using mathematical models. These models may be similar to, or the same as, the models used for determining theoretical option prices using option pricing input variables (1) - (7) discussed above. The option lot size (shares per option contract) and number of option contracts traded are typically factored into the hedging decision. The option lot size is typically defined by the options exchange when the contract is created and changed only under special circumstances, such as capital adjustments. The number of options that the trader has bought or sold may be included in the confirmation notice transmitted from the options exchange.

Accordingly, assuming a total delta hedge is desired, a trader may determine the quantity of the underlying security to be bought or sold after each option trade based on: (1) mathematical models, including price of the underlying security, (2) options lot size, and (3) number of options traded.

As noted above, the price of the underlying security may be defined in several different ways. A typical hedging response of an option trade will be to buy or sell the underlying security. Specifically, buying (selling) calls and selling (buying) puts will usually lead to selling (buying) the underlying for delta hedging. Since the trader will need to sell (buy) the

underlying to hedge the delta risk, he may be most interested in the bid (ask) price of the underlying security. While delta risk is referred to specifically, it should be understood that the automated hedging feature might be used to hedge other known risks. For example, automated hedging may be used to hedge the vega risk, the risk of a position or trade due to price changes 5 of the options arising from changes of an option's volatility.

From a trading perspective, the trader must define how and to what extent to delta hedge. Obviously, a trader must first decide whether he wants to delta hedge manually, semi- automatically, or fully automatically. In any case, he may consider two opposing dynamics: (1) speed of executing the underlying security orders, and (2) execution price of the underlying 10 security orders. Typically, a trader may choose to hedge using market orders if she is most concerned about speed of execution, or limit orders if she is most concerned about the price at which the underlying orders are executed. As described above, entering a market order will (nearly) always result in the desired quantity being executed, but at potentially unfavorable or unexpected prices. Conversely, entering a limit order will always result in executed prices 15 which meet certain criteria (i.e., greater than or equal to limit price if selling, and less than or equal to the limit price if buying), but only some or none of the desired quantity may actually be executed. Of course, other types of orders such as Fill Or Kill, and Immediate Or Cancel may be considered when designing an automated hedging feature but for brevity, will not be discussed in detail here.

The trader may assess several qualitative factors in deciding whether to semi- automatically or fully automatically hedge and, if so, whether to use market orders or limit orders. Some of the qualitative factors include: (1) the quantity of delta hedge underlying trade relative to the depth of the entire underlying market, (2) volatility of the underlying market, (3) the size of the underlying bid-ask price spread relative to the price of the underlying, and (4) 20 the amount of mental attention the trader can give toward the delta hedge trade. Different traders trading options on different underlying securities may opt for different hedging methods. Thus, in one embodiment of the automatic option trading system of the present invention, the trader may choose manual hedging, semi-automatic, or automatic hedging using 25 market orders and/or limit orders.

The fully automatic hedging software may be resident on one or more of the trader stations 230, a backend computer 220, 223, 225, or other equipment of the trader site 200. One embodiment of the fully automated hedging systems will be described in connection with Figure 6. Backend computer 220 receives option trade confirmations from exchange site 100 based on an order submitted automatically by backend computer 225. Alternatively, or in addition, backend computer 225 may receive option trade confirmations from the exchange site 100. Further, the option trade confirmations may correspond to orders submitted either automatically or manually by a trader. Moreover, the option trade confirmation may correspond to a quote submitted automatically or manually by the trader. Thus, backend computer 220 routes the trade confirmation to a trader station 230 that is associated with the automatic option trade made by the backend computer 225.

If the manual hedge feature has been selected, trader station 230 displays the appropriate hedge action based on factors previously entered by the trader. For example, the trader may see a message such as "buy 4500 shares" of the underlying security. The trader would have the responsibility of executing the hedging trade, if she desired. If the semi-automatic hedging using a limit order has been selected, trader station 230 would automatically display a dialogue box to the trader. The dialogue would include information (such as type of order, price, quantity, buy or sell) related to the potential hedging trade. If the trader is comfortable with these specifications, she may then submit this particular hedging order through actions on the keyboard, mouse, gamepad or other input device of trading station 230 (e.g., voice activation equipment). Alternatively, the trader could modify some or all of the specifications of the order before submitted the order. Of course, the trader may have the ability to cancel the dialogue box entirely if she wished not to hedge that particular option trade using the semi-automated hedging feature. If the automatic hedging using market orders feature has been selected, trader station 230 automatically submits a market order, for example to buy 4500 shares at the market prices, to exchange site 700 via backend computer 223. At the exchange site 700, the market order will (nearly) always be immediately filled by buying 4500 shares, albeit at a potentially unexpected or undesirable average price for those shares.

If automatic hedging using a limit order has been selected, trader station 230 automatically submits an order, for example to buy 4500 shares at a price of 68.05, to exchange

site 700 via backend computer 223. The default limit price specified in the semi-automated and fully automated hedging orders may depend on the current underlying market and/or parameters pre-defined by the trader. For instance, the trader may define the price to use when buying the underlying security with semi- or fully automatic hedging as, perhaps, either: (1) current ask 5 price, since she is buying, (2) average of current bid and ask price, (3) last traded price, or (4) the ask price at which the automated option trade was triggered, regardless of the current underlying prices. Depending on the market conditions, exchange site 700 may not be able to match the limit order immediately, if ever. Exchanges typically enable the trader to modify or delete 10 partially matched or completely unmatched limit orders. In some cases, the exchange site through which the underlying security is traded may depend on the option traded. For example, both futures of an equity index (e.g., Standard and Poor's 500) and options on the same equity index may be traded through exchange site 100. A stock may be traded through exchange site 700, but options for the stock may be traded through exchange site 100. In such a case, the system configurations at either the trader station 230 performing hedging or other equipment at 15 trader site 200 must ensure that hedge orders are routed to the appropriate exchange.

While the above-embodiments have been described in terms of arrays or tables, it should be understood that data may include or be maintained in other organizational memory constructs consistent with the present invention, for example, linked lists, trees, heaps, hash lists, or some combination, or any other data structure or combinations of data structures useful in 20 implementing a search algorithm. In addition, the trader site 200 is described as submitting orders to the exchange site 100 using the automated trading system. However, the trader site 200 may submit its "order" in the form of a quote to the exchange site, where the bid (ask) price of the quote corresponds to the theoretical buy (sell) price if, say, the trader wanted to buy (sell) the item.

It will be apparent to those skilled in the art that various modifications and variations can 25 be made without departing from the scope or spirit of the invention. For example, the present invention may be applied in areas other than electronically-traded securities, for example, the purchase and/or sale of goods or services, contests, auctions, and other applications in which fast, accurate responses are desirable. Other embodiments of the invention will be apparent to those 30 skilled in the art from consideration of the specification and practice of the invention disclosed

PATENT  
ATTORNEY DOCKET NO. 048289-5003

herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

卷之三